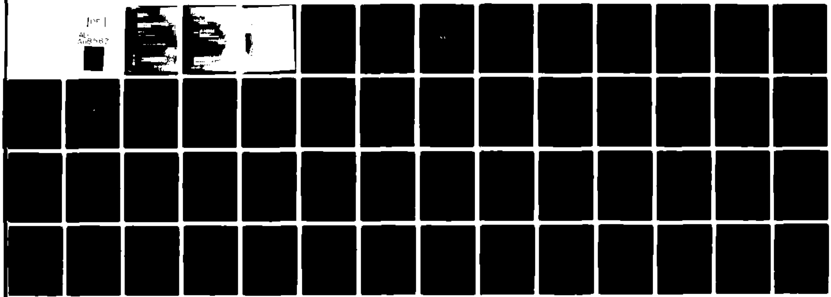
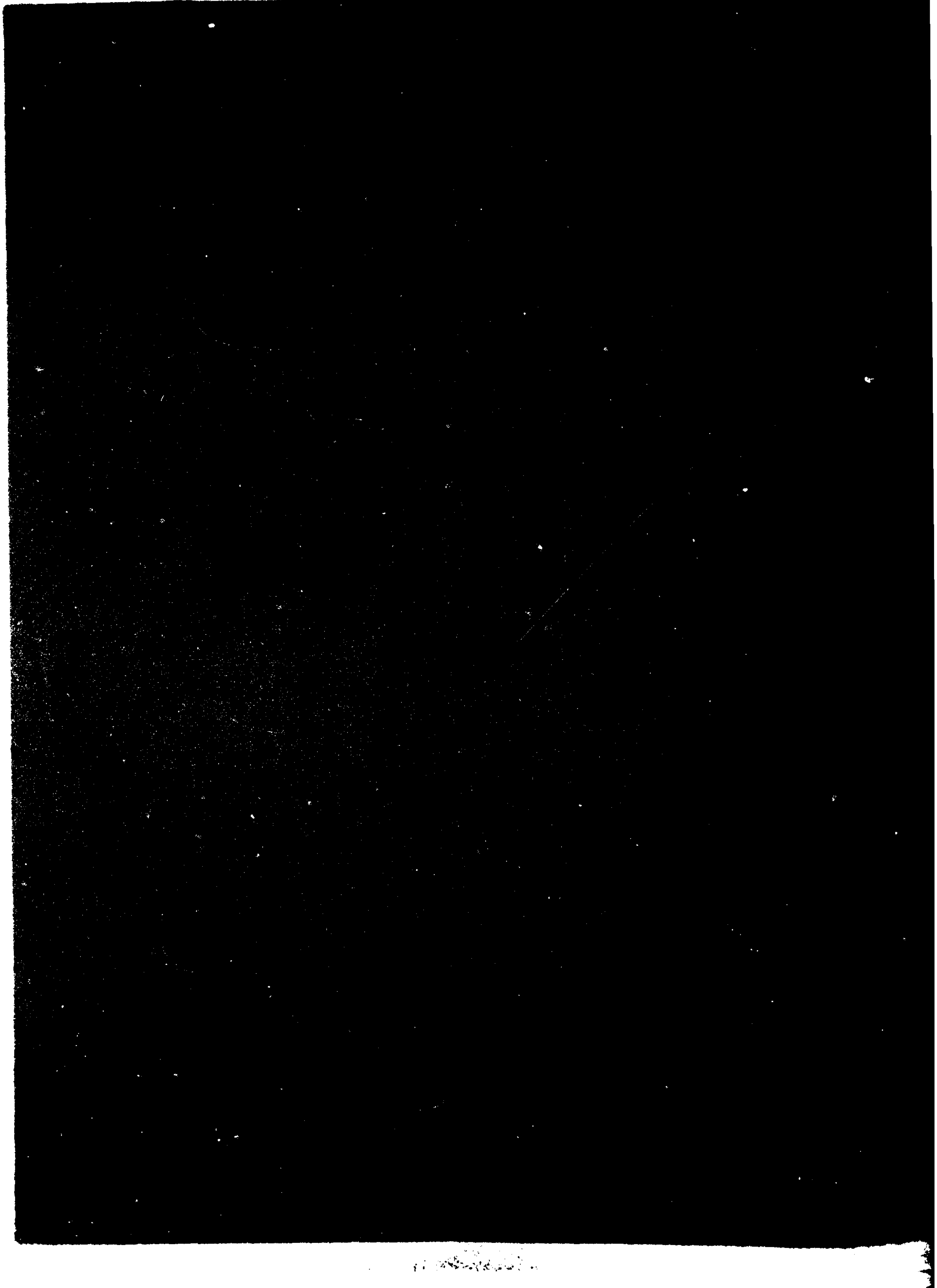
AD-A118 567 COMARCO INC ANAHIEM CA F/G 9/2
COMPUTER AIDED MISSION PLANNING SYSTEM (CAMPS).(U)
JUN 82 D BAILEY, D FRYE, W WOODBURY F30602-79-C-0230

UNCLASSIFIED RADC-TR-82-148 NL



END
DATE
FILMED
9-82
DTIC



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-82-148	2. GOVT ACCESSION NO. AD-A118567	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER AIDED MISSION PLANNING SYSTEM (CAMPS)		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1 Jul 79 - 1 Oct 81
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) D. Bailey D. Frye W. Woodbury		8. CONTRACT OR GRANT NUMBER(s) F30602-79-C-0230
9. PERFORMING ORGANIZATION NAME AND ADDRESS Comarco, Inc. 150 West Cerritos Anaheim CA 92805		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63789F 23150302
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDE) Griffiss AFB NY 13441		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 54
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Fred Haritatos (IRDE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Algorithms
computer programs
mission planning
penetration analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Algorithms and computer programs were developed to automate a wing/squadron level capability for flight planning and penetration analysis for the F-4E and F-111F; and weapons planning for the F-4E with a limited set of weapons and release conditions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-82-148	2. GOVT ACCESSION NO. AD-A118567	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER AIDED MISSION PLANNING SYSTEM (CAMPS)		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1 Jul 79 - 1 Oct 81
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) D. Bailey, D. Frye, W. Woodbury		8. CONTRACT OR GRANT NUMBER(s) F30602-79-C-0230
9. PERFORMING ORGANIZATION NAME AND ADDRESS Comarco, Inc. 150 West Cerritos Anaheim CA 92805		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63789F 23150302
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDE) Griffiss AFB NY 13441		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 54
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Fred Haritatos (IRDE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Algorithms
computer programs
mission planning
penetration analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Algorithms and computer programs were developed to automate a wing/squadron level capability for flight planning and penetration analysis for the F-4E and F-111F; and weapons planning for the F-4E with a limited set of weapons and release conditions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLANK PAGE

Evaluation

This effort was undertaken to assist the Tactical Air Force (TAF) commands refine their requirements for an operational Computer Aided Mission Planning System (CAMPS). Operational mission planning concepts were developed and implemented on experimental, testbed equipment and user feedback solicited. The significant aspect of this program was the location and maintenance support of testbed equipment at several tactical fighter units. This allowed the users to have daily contact with computer equipment and assess the benefits of automation to time-consuming, error-prove mission planning procedures.

Fred Haritatos

FRED HARITATOS
Project Engineer

FOREWORD

The CAMPS Final Technical Report has been prepared for RADC by COMARCO in response to CDRL data sequence number B007. The CAMPS software is described as it operates in final contract release 4.1.

COMARCO developed algorithms and computer programs to automate a wing/squadron level capability for flight planning and penetration analysis for the F-4E (with slats) and the F-111F; and for weapons planning for the F-4E with a limited set of weapons and release conditions.

This work is performed for the U.S. Air Force at the direction of Mr. Fred Haritatos of Rome Air Development Center under Contract No. F30602-79-C-0230.

SUMMARY

This CAMPS Final Technical Report has been prepared for RADC by COMARCO in response to CDRL data sequence number B007. This technical report describes CAMPS software release 4.1

CAMPS represents an evolution of mission planning systems. COMARCO has enhanced, augmented, and refined the CAMPS software in a series of releases aimed at providing automated assistance in conducting penetration analysis, flight planning, and weapons planning. The highly-automated, user-friendly CAMPS system enables rapid and accurate flight planning and penetration analysis for the F-4E (with slats) and the F-111F, and weapons planning for the F-4E.

The CAMPS system can be used for mission planning anywhere in the world. Digitizer, terrain elevation data bases for unmapped (DMA) areas are readily created by CAMPS. Additionally, CAMPS accepts either LCC or UTM charts for lat/long conversion to x,y coordinates for display of the mission area on the CRT. Effects of terrain masking on the defensive radars are evaluated and the probability of aircraft damage and the display template (COVERAGE or DANGER areas) are also computed.

CAMPS' enhanced flight planning capabilities enable the mission planner to display distance traveled, time, and fuel consumed/remaining for each leg and route. The final mission route display includes curves at the turn points in the flight path. In addition, CAMPS effectively automates the F-4E weapon delivery process presented in the -34 flight manual. This enables the mission planner to quickly plan his weapon delivery either as an independently executed program or in conjunction with flight planning.

TABLE OF CONTENTS

LIST OF TABLES

BLANK PAGE

1. INTRODUCTION

1.1 Purpose

The CAMPS Final Technical Report has been prepared for RADC by COMARCO in response to CDRL data sequence number B007. This report describes COMARCO's technical development efforts for CAMPS.

1.2 Scope

This report describes CAMPS software release 4.1.

1.3 Applicable Documents

a. COMARCO, Inc., CAMPS SYSTEM/SUBSYSTEM SPECIFICATION (September 1981)
b. COMARCO, Inc., EPASS PROGRAM MAINTENANCE MANUAL (March 1980)
c. COMARCO, Inc., CAMPS FUNCTIONAL DESCRIPTION WEAPON DELIVERY REQUIREMENTS
d. COMARCO, Inc., Internal Memorandum: CAMPS A7E FUEL, TIME, AND DISTANCE NOMOGRAMS, POLYNOMIAL MODEL COEFFICIENTS DATA, AND PLOTS SHOWING QUALITY OF THE POLYNOMIAL APPROXIMATIONS (December 1981)
e. COMARCO, Inc., CAMPS PROGRAM MAINTENANCE MANUAL (March 1982)

2. CAMPS SYSTEM DESCRIPTION

The Computer Aided Mission Planning System (CAMPS) provides a wing/squadron level capability for penetration analysis which includes flight planning and weapons planning. This document describes software release 4.1 of CAMPS. CAMPS provides for rapid flight plan construction, including the production of a FORM 691 flight plan, a FORM 97 sequence table, and a weapon release card. Flight planning is provided for both the F-4E with slats and the F-111F over their complete flight envelopes. Weapons planning is provided for a limited set of weapons and release conditions for the F-4E. The CAMPS software consists of seven programs, two of which are programs used day-to-day by aircrews for flight planning and weapon release planning, and five of which are used to generate and manipulate data bases.

The CAMPS stand-alone hardware suite consists of a user station (a color CRT, digitizer board, and typewriter printer) and a computer station (a processor, disk storage, and black and white CRT). The flight planner interacts with CAMPS by making data entries on the keyboard or on standard aeronautical charts and a data menu which are mounted on the digitizer board. Empty boxes on the menu are provided for growth.

In the normal mode of operation some data will already be stored in the computer: standard configuration loads (SCLs), threat types, threat locations (EOB), digitized terrain elevation data, aircraft performance data, MOB location, prominent geographic features and reference points, delta points, TACANs, sensitive areas, and the standard data table. The flight planner will first align the chart on the digitizer board (if necessary), then select a display of the threat environment. This display includes masking effects on the threat due to terrain. Next the planner selects an SCL, modifying it if necessary. By designating turn points on the chart or keyboard, the planner will then lay in a flight route. Points are selected on the chart by placing the cross-hairs of a handheld cursor (mouse) at a potential turn point on the map. A flashing cross then appears on the CRT at a corresponding point. A different potential turn point can be designated by moving the cursor, or the existing potential turn point can be entered (confirmed) as the next turn point by selecting "ACCEPT POINT" on the menu (or by touching the "7" button on the digitizer cursor). This try-before-selecting turn points method allows the planner to coordinate the flight plan, the terrain, and the threat environment while creating a flight route. Delta points and TACANs can also be used to select turn points, and turn point positions can be entered from the keyboard.

By entering such points the planner specifies route start point, turn points, IPs (initialization point), targets, route end point, external fuel tank jettison point, and offset points. Aircraft speeds, altitudes, turn radius, and wing sweep (F-111F only) can also be specified as the route is constructed. At this point in the process a complete flight plan exists; therefore the FORM 691 and FORM 97 data can be displayed on the CRT and/or printed. If time permits, the planner can refine his flight path using the INSERT, CHANGE, and DELETE commands on the menu or by using the "edit flight plan" feature. Terrain-masked threat interaction with the route can be displayed at any of three altitudes (currently 200, 500, or 1000 feet AGL).

The weapons delivery planning program provides the CAMPS user with the capability of planning his weapons delivery maneuver and of producing a Weapon Delivery Card as a final product. This program can be initiated in one of two ways: as a part of CAMPS mission planning, or as an independently executed program. When initiated as an adjunct to flight planning, data from the user's flight plan will be utilized in his weapon delivery planning. The program replicates the basic weapon delivery process presented in the -34 flight manual. The user specifies the dive angle, aircraft velocity, release altitude, and other parameters and CAMPS determines the mil lead setting, wind correction factors, time-sequenced pipper positions, and other relevant weapon delivery data.

3. WORK ACCOMPLISHED

Sections 4 through 11 describe the algorithms and computer programs developed by COMARCO to implement the capabilities of CAMPS release 4.1.

4. FUEL CALCULATIONS

4.1 Weight and Drag Effects

4.1.1 Description

Weight and drag values are provided for each leg of a route. These values reflect changes that result from fuel consumption, weapons release, and/or fuel tank release. The user inputs stores (SCLs) and current fuel for the route; the system then determines and displays dropable, non-dropable, and total weight (F-4E, F-111F) and dropable, non-dropable, and total drag (F-4E) at the outset of route planning or at store recall. F-111F drag values are dependent on user input of wing sweep for each leg of the route.

4.1.2 Implementation

4.1.2.1 Fuel Consumed

The weight of fuel consumed is subtracted from the total aircraft weight at the end of each leg.

4.1.2.2 Weight and Drag

The F-4E flight manual was utilized to create a file of dropable and non-dropable weight and drag values indexed by stores. CAMPS accesses this file by either a weapons or fuel tank category to obtain weight and drag values for the F-4E. The F-111F flight manual was utilized to create a table of dropable and non-dropable weight and drag values indexed by wing sweep, store, and MACH number. Using the inputs of wing sweep and stores loaded at the start of each leg, CAMPS performs a table lookup to output the weight and drag values for the F-111F.

4.1.2.3 Weapons Release

The user-specified weapons release point on a route is the same as the target point on a route. The target point, by nature of CAMPS, serves as the ending point of one leg and the starting point of the next leg. From the SCL formed at the weapons release point, new store inputs are used in the F-4E data-file (weapons category) or in the F-111F table-lookup. The outputs provided are new weight and drag values which are then used in the calculations for the next leg of the route. Weapons are released equally over the number of targets in the route.

4.1.2.4 Fuel Tank Release

CAMPS assumes that external fuel is used before internal fuel. All fuel tanks are jettisoned at a fuel tank release point. The fuel tank release point is a user specified point on a leg of the route. From the SCL formed at this point, new store inputs are used to access the F-4E weight and drag data file (fuel tanks category) or the F-111F weight and drag table lookup. The outputs provided are the new weight and drag values. Because of the nature of the fuel calculations/fuel performance routines, CAMPS generates a point on the route to insert these new weight and drag values according to the following criteria:

a. If the flight mode at the fuel tank release (FTR) point is the cruise mode, then for one-half the distance in cruise the old weight and drag values are used, and for one-half the distance the new weight and drag values are used.

b. If the flight mode at the FTR point is either the climb mode or descent mode, the new weight and drag values are used for the entire flight mode distance.

4.2 Fuel Performance Routines/Flight Mode Determination

4.2.1 Description

The flight mode(s) for each leg of a route are determined from analysis of the altitudes specified at the start point and end point of the leg. User inputs of beginning and ending altitudes are processed to produce outputs of cruise, climb, or descent flight mode(s).

4.2.2 Implementation

CAMPS accepts start and end point altitude data input by the user in either AGL or MSL. All calculations are performed on MSL; therefore CAMPS converts AGL values to MSL values after accessing the terrain data base. From the user inputs, the start point altitude is subtracted from the end point altitude. If the result is greater than A Climb, the flight mode is climb; less than -A Descent, the flight mode is descent; otherwise, the flight mode is cruise. Currently A Climb and -A Descent are set at 10,000 feet; these values will change to 5,000 feet for release 4.2.

Inputs of flight mode, beginning and ending speed, and total distance of the leg are used in the fuel performance routines to determine the distance traveled in each flight mode and complementary modes for each leg:

- o Climb - Determine climb distance; remaining distance in cruise mode.
- o Descent - Determine descent distance; remaining distance is the cruise mode which occurs before the descent.
- o Cruise - In cruise mode for entire leg.

If the leg is too short for a descent or a climb, CAMPS displays an error message to the user.

4.3 Fuel Performance Routines

4.3.1 Description

CAMPS provides fuel consumed/remaining, distance, and time values for three flight modes (cruise/climb/descent) for the F-4E and the F-111F. The user inputs speed, altitude, and wing sweep. CAMPS provides the flight mode, weight, drag, and distance (cruise only) values to the fuel performance routines. Outputs of fuel consumed/remaining, time, and distance are calculated and displayed for the current leg and for the total route.

4.3.2 Implementation

4.3.2.1 F-4E (Climb/Descent Modes); F-111F (Cruise/Climb/Descent Modes)

A number of development efforts were necessary to design the fuel performance routines. The fuel performance routine for each flight mode (except F-4E cruise) is a series of polynomials. The order of progression in this set of polynimials mirrors the pilot's manual method of reading the performance nomograms in the flight manuals. COMARCO utilizes the TALOS digitizer to convert the performance nomograms to polynomials. Specific polynomials and their coefficients are well documented in the software. COMARCO designed the program CRVFIT to determine the coefficients for these polynomials, and the program PTTEST to determine their accuracy/reliability.

The cruise flight mode assumes that a cruise mach, pressure altitude, and gross weight and drag are specified. Only fuel consumption rate data results from this mode since the performance driver calculates leg time and distance in the cruise mode. The descent mode provides the time, distance, and fuel data which result from maintaining a constant CAS. The CAS is determined as a function of given values of gross weight and drag count.

Both climb and descent modes are specified with respect to sea level. Therefore, if the climb/descent begins or ends at a non-sea level altitude, two sets of time, distance, and fuel values for each mode must be determined. The difference in the two sets of values determines the correct results.

4.3.2.2 F-4E (Cruise Mode)

A similar curve-fit multiple regression and analysis program which operates on tabular data from the F-4E flight manual was utilized to determine the coefficients for the F-4E cruise polynomial (Table 4.3.2.2-1).

5. THREAT MODELING

5.1 Description

Subroutine C MODEL computes the probability of aircraft damage (PD) from threat activity, and the effectiveness of threat types as a function of aircraft speed, altitude, and direction of flight, and of the ECM used.

5.2 Implementation

The algorithm compares the criteria with the maximum possible damage. Inputs are adjusted for ingress and egress vulnerability, low altitude effects, and the ECM used. PD is computed as a function of speed, altitude, and direction of flight.

The Fortran code in Table 5.2-1 provides the algorithm.

6. TERRAIN GENERATION

6.1 Description

In order to determine aircraft altitude during fuel performance computations, CAMPS requires digitized terrain elevation data base. The terrain generation program was designed to permit the creation of a suitable data base anywhere in the world.

Table 4.3.2.2-1 F-4E (with slats) Cruise Implementation

$$FR = C_1V + C_2D + C_3ALT + C_4GW + C_5$$

Where:

FR = Fuel rate: 10,000 lbs/hr

C = Coefficient

V = Speed in knots

D = Drag

ALT = Altitude in feet (MSL)

GW = Gross weight

Table 5.2-1(a) Threat Modeling Implementation

```
THETA = 0.
ALT=(AALT)*XKFTNM
ZMAX=ZMAXX
ROUT=ROUTX

 FOR PDPLOT PROGRAM CHECK VIEW:  1=PLAN  2=SIDE
IF (IVIEW.EQ.2)GO TO 5
IF (X.EQ.0..AND.Y.EQ.0) GO TO 5
THETA = ASIN (X/SQRT(X**2 + Y**2))
COSTHT = COS(THETA)
IF (Y.LT. 0.0) COSTHT = -COSTHT

R      = SQRT (X**2 + Y**2 + ALT**2)
PHI    = ASIN(ALT/R)
COSPHI= COS(PHI)
TEST FOR ALTITUDE  CUTOFF
IF (PHI.L.PHIMIN) GO TO 100
IF (PHI.GE.PHIMAX) GO  TO 100
IZ=ZMAX
  EXTRACT ZCRIT FROM FRACTIONAL DIGITS OF ZMAX.
ZCRIT=((ZMAX-IZ)*10000.)/6080
ZMAX=ZMAX/6080.
DELTAR=1.25*ZCRIT
PHICRT=ATAN2(ZCRIT,ROUT)
SIGMA=1. -2.84E-5*ALT + 2.508E-10*ALT**2
POTR=ROUT +DELTAR
RSL=POTR/SQRT(SIGMA)
```

Table 5.2-1(b) Threat Modeling Implementation

```
      IF (PHI.LT.PHICRT) GO TO 10
      REF=(((ROUT*ZMAX)**3)/((ZMAX*COSPHI)**3 +
 *        (ROUT*SIN(PHI))**3))**POWER
      GO TO 20
      REF=SQRT((RSL+((ALT/ZCRIT)*(ROUT-RSL)))**2+
 *        ALT**2)*(ABS((PHI-PHIMIN)/(PHICRT-PHIMIN))
          **POWER2)
 *    CONTINUE
      IF(REF.LT.ALT) GO TO 100
      HORRNG=SQRT(REF**2-ALT**2)
      HORRNG-HORRNG*(1.+SPEEDA*COSTHT*COSPHI/SPEEDM)
      REF-SQRT(HORRNG**2+ALT**2)
      IF(REF.EQ.0.) GO TO 100
      IF (R/REF.LE.0.OR.R/REF.GE.1) GO TO 100
      IF (R.LE. RIN*(1.0+SPEEDA*COSTHT*COSPHI/SPEED M))GO TO 100
      IF(R/REF.LE.L) REQ=.5*(R/REF)/L
      IF(R/REF.GT.L) REQ=1.-(1.-R/REF)/(2.-2.*L)
      ABSINT = ABS(SIN(THETA))
      ABSINP = ABS(SIN(PHI))
      PDREAL = PINIT*(4.0*(REQ-REQ**2))**C1 *
 A              (1.0-K2*(SPEEDA*ABSINT/R+SPEEDA*ABSINP/R)) *
 B              (1.0-K3*(ABSINT**2 + ABSINP**2))
      ADJUST FOR VULNERABILITY, INGRESS/EGRESS
```

Table 5.2-1(c) Threat Modeling Implementation

```
ALPHA=SQRT(THETA**2+PHI**2)
IF (Y.LT. 0.0) ALPHA = 3.1415927 - ALPHA
PDREAL = PDREAL*(1.0-K7*EXP(-EXP(-K8/3.2 - ALPHA) + K9)))
ADJUST FOR LOW ELEVATION (ALT) EFFECTS. .0548=3.1DEG
IF(PHI.LE.PHILOW) PDREAL-PDREAL*(1.-K1*COS(PI(5)/PHILOW**PHI))

ADJUST FOR ECM = SP
IF (ECM.EQ.2) PDREAL = PDREAL*(1.0-K4*COS(THETA)*COSPHI)
ADJUST FOR ECM = SOJ
IF (ECM.EQ.3) PDREAL = PDREAL*(1.0-K10*(5.-(RANGE/ROUT))
*/(6.-(RANGE/ROUT)))
ADJUST FOR ECM = EJ
IF (ECM.EQ.4) PDREAL = PDREAL*(1.0-K5*COS(THETA)*COSPHI)
IF (ECM.EQ.4) PDREAL=PDREAL*(1.0-K5)
ADJUST FOR ECM = CHAFF
IF (ECM.EQ.5) PDREAL = PDREAL*(1.0-K6*EXP(-SPEEDA/500.0))
PD = PDREAL * 100.0
IF (PD.GT.99) PD = 99
IF (PD.LT.0) PD = 0
RETURN
PD = 0
RETURN
```

Table 5.2-1(d) Threat Modeling Implementation

where:

ROUT = STATIC OUTER BOUNDARY (NM)

RIN = STATIC INNER BOUNDARY (NM)

ZMAX.ZCRIT = MAX ALTI .CRIT ALTI (FEET HUNDREDS OF FEET)

ZCRIT = ALT BELOW WHICH MAX RNG DECREASES DUR TO MISSILE ENERGY LIMITS.

PHIMIN = MIN TRACKING ELEVATION ANGLE (DEG)

PINIT = INIT PD, ON FRONT, BEFORE OTHER EFFECTS

THETA = AZIMUTH ANGLE TO AIRCRAFT (POSITIVE INGRESS)

PHI = ELEVATION ANGLE TO AIRCRAFT

SPEEDA = AIRCRAFT VELOCITY

SPEEDM = MISSILE VELOCITY

DELTA = DYNAMIC RANGE CORRECTION

L = LOCATION OF PD PEAK ($0<L<1$)

R = STATIC SLANT RANGE

C1 = KURTOSIS FACTOR

[C1 > 1 PEAKIER THAN PARABOLA]

[C1 $<$ 1 FLATTER THAN PARABOLA]

K1 = LOW ELEV (ALT) DEGRADATION FACTOR

K2 = SLEW RATE FACTOR

K3 = BURST ANGLE FACT

K4 = SPJ ECM MULT

K5 = EJ ECM MULT

K6 = CORR CHAFF ECM MULT

K7 = FRONT-SIDE/OVERHEAD-REAR SCALE FACTOR

K8 = FRONT-SIDE/OVERHEAD-REAR SLOPE FACTOR

Table 5.2-1(e) Threat Modeling Implementation

K9	=	F-S-R LOCA FACT
K10	=	SOJ ECM MULT
X	=	OFF-TRACK
Y	=	DOWN-TRACK
ALT	=	AIRCRAFT ALTITUDE (IN NM)

Aeronautical charts are mounted on the digitizer and terrain contours are mapped with the digitizer cursor. The program instructs the user throughout the multistepped process, giving the user opportunities to test and correct data entry problems at several intermediate steps.

Points entered through the digitizer are stored in an open data file. When all points have been input and tested, the program fills in all the remaining locations by using an averaging algorithm. These points are entered into another data file, which is compared with the first. All points are checked for compatibility with adjacent points.

6.2 Implementation

6.2.1 Subroutine TERPO

This subroutine takes the elevations stored during digitizer input and interpolates between values to fill all the cells of the terrain data file. The algorithm sums each elevation with adjacent point elevations and averages these values to determine intermediate values.

6.2.2 Subroutine TERPE

This subroutine fills input elevations into a temporary data base. When necessary, this program will fill in with a straight line interpolation between input points to fill all cells of the data base. The algorithm develops a smooth transition between user input points. It interpolates a straight line between two input elevations and fills the cells (1,000/inch) between points with this data.

7. DOMINANT TERRAIN MASKING FUNCTION DETERMINATION

7.1 Description

The problem of describing the effect that terrain has on the capability of threat sites is handled through a number of algorithms. These algorithms account for elevation of site, altitude of strike force, threat angle of sight, surrounding elevations, and true ground distance. By comparing the tangents of generated angles (between the ground range and the elevation) incrementally along the path of the strike force, masking can be determined. When the tangent of the elevation angle is less than the tangent associated with the ground range, the strike force is masked.

7.2 Implementation

Subroutine MASK builds and writes the dominant masking function (DMF) to the data files. The algorithm used for this purpose establishes the static threat boundary. A circle is drawn to represent this threat boundary (CPAGL). The ground range is then determined between the threat and strike force and the difference in altitude is calculated. The azimuth angle is computed; when the tangent of this angle is compared to the tangent of the site's highest possible angle (allowing for elevation of site), the strike force mask condition is determined.

The Fortran code in Table 7.2-1 provides the algorithm.

Table 7.2-1 Dominant Terrain Masking Function Determination Implementation

```
X  = RNG * COSRAD
Y  = RNG * SINRAD
IX = 1.5 + (XLON - EXTENT(2)) * 4
IX = 1.5 + (EXTENT(3) - YLAT) * 4
! ELEVATION OF TERRAIN IN NAUTICAL MILES
THETA = 1.5707963 - (RNG * PRENM)  !  ANGLE THETA
HYPOTN = RENM + HT  ! HYPOTENUSE
RX = HYPOTN * COS(THETA)  ! X COMPONENT OF TRIANGLE
RY = HYPOTN * SIN(THETA)  ! Y COMPONENT OF TRIANGLE
TA = (RY - (RENM + ELVNM)) /RX ! TANGENT OF LINE OF SITE
LARGER ANGLE FOUND
```

8. COVERAGE AND DANGER TEMPLATE GENERATION

8.1 Description

Subroutine CPAGL creates the area and circle of coverage in AGL.

8.2 Implementation

Maximum terrain increments are checked against the tangent of the angle range data. The maximum angle for each radial point from the threat is compared to the elevations from the data base files in order to determine which points are shadowed from the site.

9. CHART TO INTERNAL MAP COORDINATE CONVERSION

9.1 Description

In order to be able to use any map available from any operational theater, a procedure was necessary to define the map playing area in a manner useful to the CAMPS program. This section will cover the concepts and software involved in projecting points onto a graphics screen, when those points are input either from a digitizer "mouse" or from lat/long keyboard entries.

In the tablet mode, using the digitizer board and mouse, the user digitizes the SW and NE corners of the map to be used. CAMPS then uses subroutine CTABLT to convert these coordinates to subject space coordinates.
In the keyboard mode the procedure is the same, except that the digitizer is not used. The user inputs lat/long directly from the keyboard in degrees/minutes/seconds. CAMPS uses subroutine DIRLCC or DIRUTM to convert these to subject space coordinates. CAMPS then uses subroutine PRELIM and EXTNTS to set up the necessary mapping constants for x,y to lat/long conversions, and to determine the best real space extents (subject space), for mapping onto the plotting spaces (object space) while retaining real space aspect ratios.

9.1.1 Object Space

The object space is the region of the screen within which plotting is to take place. The screen (real space) is addressed by units called rasters. The RAMTEK graphics screen is 512 rasters by 256 rasters. Ideally these rasters would be equally spaced in the vertical and horizontal direction, but on the RAMTEK the spacing is 1 raster in y to 1.5 rasters in x. The extent of any space is defined by giving the coordinates of the bottom left and upper right corners of the space. The entire plottable area has been assigned the object space extent (0,0) (4095,3071) by definition. In some instances it will be desired to plot a figure within some subregion of the object space such as (X1, Y1) (X2, Y2). In this case, the active object space is redefined from the default condition using subroutine OBJCTG, and all subsequent plotting will be done in this region.

9.1.2 Subject Space

The subject space assigns the correspondence between the object space and the coordinates of the space to be plotted. All plot requests are made in subject space units. As an example, assume that a graph of probability versus time is to be drawn within the active object space. Assume further that time ranges from -30. to 90. seconds and that probability ranges from 0. to 1. In this case the subject space would be assigned the extent (-30., 0.) (90., 1.) To plot a line from subject space coordinate (-30., .4) to (10., .2), the program maps the coordinates from the subject space to the object space to the real device space. A result of the subject space/object space procedure is that it is a simple matter to translate, shrink, or expand the plot as desired.

9.2 Subroutine CTABLT

9.2.1 Description

The function of the subroutine is to convert a digitizer coordinate set (x,y) to a subject space coordinate set in nautical miles.

9.2.2 Implementation

This conversion is accomplished by multiplying the change in distance from the lower left extent coordinates by the cosine and sine of the map tilt angle. Utilizing the equations in Table 9.2.2-1 conversion factors are then used to derive NMI from digitier units in thousandths of an inch.

9.3 Subroutine PRELIM

9.3.1 Description

This subroutine performs the preliminary calculations which are common to both Universal Transverse Mercator (UTM) and Lambert Conformal Conic (LCC) mapping systems conversions. This subroutine may also be used to calculate the distance between two lat/long coordinates.

9.3.2 Implementation

Constants are obtained from a spheroid constant table (one of three) and used to calculate an LCC convergence constant, equator radius constant, and a base latitude radius. When using the digitizer, input adjustment is also made for the map tilt angle on the digitizer board. This is accomplished using the algorithms in Table 9.3.2-1.

9.4 Subroutine DIRLCC

9.4.1 Description

When a map using Lambert Conformal Conic (LCC) projection is used, this subroutine performs the direct conversion from LCC lat/long to x,y coordinates.

Table 9.2.2-1 CTABLT Implementation

$$RLON = (RX-XA)\ \cos\theta + (RY-YA)\ \text{sine}\ \theta\ XC$$

and:

$$RLAT = (XA-RX)\ \text{sine}\theta + (RY-YA)\ \cos\theta\ YC$$

Where:

RLAT, RLON	= coordinate converted to subject space coordinates in NMI
RX, RY	= digitizer coordinate (10^{-3}/in)
XA, YA	= digitizer lower left extent coordinate (10^{-3}/in)
Cos θ	= cosine map tilt angle
Sine θ	= sine map tilt angle
θ	= map tilt angle
XC, YC	= conversion factors for digitizer to NMI in X and Y (NMI/10^{-3}/in)

Table 9.3.2-1 PRELIM Implementation

Given:

SROIDC - Spheroid constant table:

Row 1 - a, b, e for International Spheroid.

Row 2 - a, b, e for Clarke 1866 Spheroid.

Row 2 - a, b, e for Bessel Spheroid.

WORLD - World area table, defining areas of the world that correspond to the above spheroid. If the playing area or extent does not fall within one of the given world areas, the default condition is International Spheroid.

Determine a, b, e depending on the world area of interest. Then calculate the following constants:

$$e' = \sqrt{a^2+b^2/b}$$

$$w = (a-b)/(a+b)$$

$$L = \frac{\log|\cos\phi_1| - \log|\cos\phi_2| + \log N_1 - \log N_2}{\log|\tan(\frac{Z_1}{2})| - \log|\tan(\frac{Z_2}{2})|}$$

$$K = \frac{N_1|\cos\phi_1|}{L|\tan^L(\frac{Z_1}{2})|} = \frac{N_2|\cos\phi_2|}{L|\tan^L(\frac{Z_2}{2})|}$$

$$RB = K|\tan^L(\frac{Z_1}{2})|$$

where:

$$N_1 = \frac{a}{\sqrt{1-e^2\sin^2\phi_1}} \qquad Z_1 = 90^o - \phi_1$$

$$N_2 = \frac{a}{\sqrt{1-e^2\sin^2\phi_2}} \qquad Z_2 = 90^o - \phi_2$$

9.4.2 Implementation

Equations using a map convergence constant multiplied by a change in latitude or longitude determine the angle of convergence. The sine and cosine of this angle are then multiplied by the radius of the latitude to obtain x,y coordinates as shown in Table 9.4.2-1.

9.5 Subroutine DIRUTM

9.5.1 Description

The function of this subroutine is to perform direct latitude/longitude to x,y coordinate conversions for the Universal Transverse Mercator (UTM) projections.

9.5.2 Implementation

This is implemented by using the eccentricity and axis of the spheroid and the meridional arc length from the equator to the latitude in the equations which are shown in Table 9.5.2-1.

9.6 Subroutine INVLCC

9.6.1 Description

This subroutine is used to convert (x,y) coordinates into Lambert Conformal Conic (LCC) lat/long coordinates.

9.6.2 Implementation

Using the radius of the earth at a base latitude, a convergence factor, a constant radius from spheroid tables, and given standard parallels, perform the desired conversion. Implementation is shown in Table 9.6.2-1.

9.7 Subroutine INVUTM

9.7.1 Description

This subroutine is used to perform inverse Universal Transverse Mercator (UTM) conversion from (x,y) to lat/long.

9.7.2 Implementation

The axes and eccentricity of the spheroid are used with the central meridian to convert the coordinates. Implementation is shown in Table 9.7.2-1.

9.8 Subroutine EXTNTS

9.8.1 Description

This subroutine determines the best usage of the graphics screen to display the map extents requested. The entire screen area is used and real space aspect ratios are retained.

Table 9.4.2-1 (a) DIRLCC Implementation

Direct conversion (LAT/LONG to X/Y):

$X = R \sin \theta$

$Y = RB - R \cos \theta \qquad \theta = L \cdot (\lambda - \lambda_o)$

where: L = Map convergence constant

λ = Longitude

λ_o = Average longitude

RB = Radius at the base latitude.

R = Radius at the input latitude.

$R = K \tan^L (Z/2)$ and $Z = \frac{\pi}{2} \quad |\phi|$

where: K = Radius at the equator

Z = Conformal colatitude

θ = Latitude.

If in tablet mode, also calculate the following:

Map tilt angle $\theta = \frac{\pi}{2} - \tan^{-I} (\frac{\Delta XD}{\Delta XD}) - \tan^2 (\frac{\Delta XW}{\Delta YW})$

where: ΔXD, ΔYD are the spans in X, Y for the digitizer extent.

ΔXW, ΔYW are the spans in X, Y for the world area extent.

Adjust the digitizer space for the map tilt.

$\Delta XD' + \Delta XD \cdot \cos\theta + \Delta YD \cdot \sin\theta$

$\Delta YD' = \Delta YD \cdot \cos\theta - \Delta XD \cdot \sin\theta$

Set up constants for converting x/y (IN) to x/y world (NMI).

$XC = |\Delta XW/\Delta XD'| \qquad YC = |\Delta YW/\Delta YD'|$

<u>Distance Calculations</u>:

Convert $\phi_1, \lambda_1, \phi_2, \lambda_2$, to X_1, Y_1, X_2, Y_2.

Table 9.4.2-1 (b) DIRLCC Implementation

Then compute the distance between the points, where:

ϕ	= Latitude
λ	= Longitude
θ	= Map tilt angle
L	= Convergence constant for Lambert Conformal Conic map
K	= Equator radius constant
RB	= Base latitude radius
XC,YC	= Conversion factor for digitizer space to subject space in x, y.

Table 9.5.2-1 (a) DIRUTM Implementation

Given: a = Semi-major axis of spheroid.

b = Semi-minor axis of spheroid.

ε = Eccentricity of spheroid.

ε' = Second eccentricity of spheroid.

$$w = \frac{a-b}{a+b}$$

$$A' = a\{1-w + 1.25\ (w^2-w^3) + 1.265625\ (w^4-w^5)\}$$

$$B' = \{1.5(a)\ (w)\ (1 + w(\ -1+w(-.875+.859375w))))\}$$

$$C' = .9375a\{w^2-w^3+.75(w^4-w^5)\}$$

$$D' = .729166667a\{w^3-w^4+.169230769w^5\}$$

$$E' = .615234375a\{w^4-w^5\}$$

$$S = A'\ \phi-B'\ \sin2\phi+C'\sin4\phi-D'\sin6\phi+E'\sin8\phi$$

$$(I) = .9996S$$

$$(II) = \frac{.9996\nu\sin\phi\cos\phi}{S}$$

$$(III) = \left(\frac{.9996\nu\sin\phi\cos^3\phi}{24}\right)(5-\tan^2\phi+9\varepsilon'^2\cos^2\phi+4\varepsilon'^4\cos^4\phi)$$

$$(IV) = .9996\nu\cos\phi$$

$$(V) = \frac{(.9996\nu\cos^3\phi)}{6}(1-\tan^2\phi+\varepsilon'^2\cos^2\phi)$$

$$A_6 = \left(\frac{.9996(\Delta\lambda)^6\nu\sin\phi\cos^5\phi}{720}\right)(61-58\tan^2\phi+\tan^4\phi+270\varepsilon'^2\cos^2\phi-330\varepsilon'^2\sin^2\phi)$$

$$B_5 = \left(\frac{.9996(\Delta\lambda)^5\nu\cos^5\phi}{120}\right)(5-18\tan^2\phi+\tan^4\phi+14\varepsilon'^2\cos^2\phi-58\varepsilon'^2\sin^2\phi)$$

$$y = (I) + (II)\ (\Delta\lambda)^2 + (III)\ (\Delta\lambda)^4 + A_6$$

$$x' = (IV)\ (\Delta\lambda) + (V)\ (\Delta\lambda)^3+B_5$$

$$x = x' + 500{,}000$$

Table 9.5.2-1 (b) DIRUTM Implementation

Where:

ϕ	= latitude (radians)
λ	= longitude (radians)
x	= x rectangular coordinate (meters)
y	= y rectangular coordinate (meters)
s	= meridional arc length from equator to (meters)

A_6 and B_5 - negligible error terms

$\Delta\lambda$	$= \lambda - \lambda_0$ where λ_0 is longitude of central meridian

Table 9.6.2-1 INVLCC Implementation

Algorithm

Inverse Conversion (x/y to Lat/Long):

Given: L, RB, λ_o , K, X, Y

$$\lambda = \frac{\Theta}{L} + \lambda_o \quad \text{where } \Theta = \tan^{-1}\left(\frac{X}{RB-Y}\right)$$

$$\phi = \frac{\pi}{2} - 2\tan^{-1}\left(\frac{R}{K}\right)^{\frac{1}{L}} \quad \text{where } R = \frac{RB-Y}{\cos\Theta}$$

λ = Longitude

ϕ = Latitude

RB = Radius of earth at base latitude

L = Convergence factor

λ_o = Average longitude

K = Radius constant based on spheroid and standard parallels

x = Horizontal coordinate

y = Vertical coordinate.

Table 9.7.2-1 (a) INVUTM Implementation

Given: a = Semi-major axis of spheroid

b = Semi-minor axis of spheroid

ε = Eccentricity of spheroid

ε' = second eccentricity of spheroid

λ_o = central meridian

$$C_0 = 1 - \frac{3}{4}\varepsilon'^2 + \frac{39}{64}\varepsilon'^4 - \frac{133}{256}\varepsilon'^6 + \frac{7491}{16384}\varepsilon'^8$$

$$D_1 = \frac{3}{4}\varepsilon'^2 - \frac{3}{8}\varepsilon'^4 + \frac{213}{1024}\varepsilon'^6 - \frac{255}{2048}\varepsilon'^8$$

$$D_2 = \frac{21}{128}\varepsilon'^4 - \frac{21}{128}\varepsilon'^6 + \frac{1599}{12288}\varepsilon'^8$$

$$D_3 = \frac{151}{3072}\varepsilon'^6 - \frac{453}{6144}\varepsilon'^8$$

$$\gamma = \sqrt{\frac{1+\varepsilon'^2}{b}} \quad \frac{Coy}{.9996}$$

$$\phi' = \gamma + \frac{1}{2}\left(D_1 \sin 2\gamma + D_2 \sin 4\gamma + D_3 \sin 6\gamma\right)$$

$$\nu = \frac{a}{\sqrt{1-\varepsilon^2 \sin^2\phi'}}$$

$$(VII) = \frac{\tan\phi'}{2\nu^2}\left(1+\varepsilon'^2\cos^2\phi'\right)$$

$$(VIII) = \frac{\tan\phi'}{24\nu^4}\left(5+3\tan^2\phi' + 6\varepsilon'^2\cos^2\phi' - 6\varepsilon'^2\sin^2\phi' - 3\varepsilon'^4\cos^4\phi' - 9\varepsilon'^4\cos^2\phi'\sin^2\phi'\right)$$

$$D_6 = \left(\frac{x}{.9996}\right) 6 \frac{\tan\phi'}{720\nu^6}\left(61+90\tan^2\phi' + 45\tan^4\phi' + 107\varepsilon'^2\cos^2\phi' - 162\varepsilon'^2\sin^2\phi' - 45\varepsilon'^2\tan^2\phi'\sin^2\phi'\right)$$

$$\phi = \phi' - (VII)\left(\frac{x}{.9996}\right)^2 + (VIII)\left(\frac{x}{.9996}\right)^4 - D_6$$

$$(IX) = \left(\frac{\sec\phi'}{\nu}\right)$$

$$(X) = \frac{\sec'}{6\nu^3} - \left(1+2\tan^2\phi'\varepsilon'^2\cos^2\phi\right)$$

Table 9.7.2-1 (b) INVUTM Implementation

$$E_5 = \left(\frac{x}{.9996}\right) 5 \frac{\sec\phi'}{120\nu^5} (5+28\tan^2\phi+24\tan^4\phi'6\,\varepsilon'^2\cos^2\phi+8\,\varepsilon'^2\sin^2\phi)$$

$$\lambda' = (IX)\left(\frac{x}{.9996}\right) - (X)\left(\frac{x}{.9996}\right)^3 + E_5$$

$$\lambda = \lambda' + \lambda_0$$

where:

ϕ = latitude (radians)

λ = longitude (radians)

x = x rectangular coordinate (meters)

y = y rectangular coordinate (meters).

9.8.2 Implementation

The graphic screen (available field) is divided into equally spaced units; then the subject space extents input by the user aré scaled by a percentage onto the available field as shown in Table 9.8.2-1.

10. STRIKE ROUTE GENERATION

10.1 Description

From user specified points and from speed, altitude, and turn data, CAMPS displays a strike route showing curves at turn points and a rough estimate of the time and distance traveled in each leg of the route.

10.2 Implementation

Two programs are necessary for strike route generation. CAMPS uses one program (STRK2) to determine the rollout at each turn point and the distance and heading on each leg. Another program (STRIK) computes the number of points on the strike path and, using the data from STRK2, generates the turning points and the distance, time, and heading for all points except the last point.

The equations shown in Table 10.2-1 were developed to implement the turn points routine.

11. WEAPONS PLANNING

CAMPS provides the user with the capability to plan his weapons delivery and to produce, as a final product, a Weapon Delivery Card. This program can be initiated in one of two ways: as a part of CAMPS mission planning or as an independently executed program. When initiated as an adjunct to flight planning, data from the user's flight plan will be utilized in his weapon delivery planning. The program replicates the basic weapon delivery process presented in the -34 flight manual. The user specifies the dive angle, aircraft velocity, release altitude, and other parameters and CAMPS determines the mil lead setting, wind correction factors, time sequenced pipper positions, and other relevant weapon delivery data.

11.1 Ballistic Tables

11.1.1 Description

From user inputs regarding release conditions, CAMPS provides ballistics information for an F-4E dive bomb weapon delivery method:

Weapon; Suspension	Release Mode
BDU-33B/B; SUU-20/A	Single
BDU-33B/B; MER/TER	Single
MD 82; MER/TER	Ripple
MK 82; Snakeye 1; MER/TER	Ripple

Table 9.8.2-1 EXTINTS Implementation

XA, YA	Current extents LL corner
XB, YB	Current extents UR corner
OBJ	Object space extents
X_1, Y_1	New scaled extents LL
X_2, Y_2	New scaled extents UR
X min, Y min	Redefined subject space
X max, Y max	

Compare X_1 : XA ; Y_1 : YA ; and so on as necessary to obtain scale of object space to new subject space.

Y:X always (1:5)

X:Y always (1.333) (4095/3071)

WDO = $X_2 - X_1$ HTS = old input space extents (Sub 4) - new input space extents (Sub 2) - (Sub 4) - (Sub 2)

HTO = $Y_2 - Y_1$ WDS = (Sub 3) - (Sub 1)

$$RK = \frac{HTO}{HTS} \cdot \frac{WDS}{WDO}$$

Table 10.2-1 Turn Point Implementation

$$R_{NM} = .0000145\ V^2_{KTAS}$$

" 45° BANK"

$$R_{NM} = .0000145\ V^2_{KTAS} / \sqrt{15}$$

" 4g ROLLOUT"

11.1.2 Implementation

CAMPS utilizes the -34-1-2 Ballistics Tables to create a direct access file to be searched using verified flight parameters. The single release mode tables are placed in a file and interpolation is used within the table; no extrapolation is allowed. The ripple drop release mode data are computed using information from the single drop tables. These results are compared with -34-1-2 ripple tables for accuracy and reliability. User inputs of weapon, dive angle, altitude, and speed provide the necessary ballistic data for weapons planning.

11.2 Fuze Arming and Safe Escape

11.2.1 Description

Using flight plan information, CAMPS allows the pilot to compare the desired weapon release altitude with both the minimum fragment clearance altitude and the recovery altitude.

11.2.2 Implementation

CAMPS uses a direct access file containing fuze arming and safe escape data based on -34-1-1 tables and their interpolations. From user inputs CAMPS determines the minimum release altitudes for bomb fuze arming (FUZE ARMING) and the minimum release altitude for bomb fragmentation clearance (SAFE ESCAPE), assuming a 4.0G recovery is attained within two seconds after release. If necessary, CAMPS allows the user to alter the original release altitude or fuze arming delay values and to use the comparison process again to plan safe conditions for weapon release.

11.3 Altimeter Lag

11.3.1 Description

CAMPS provides altimeter lag values dependent on SPC status (on/off).

11.3.2 Implementation

The altimeter lag is determined for two conditions: SPC-on and SPC-off. CAMPS calculates an approximate derated true airspeed value and enters it in one equation if the SPC is on, and in another equation if the SPC is off. Adjustments for dive angle and true airspeed are made based on the 34-1-2 nomograms, and altimeter lag values are generated for SPC-on and SPC-off.

The Fortran code in Table 11.3.2-1 provides the algorithm used.

11.4 Altitude Loss During Pull-up

11.4.1 Description

The altitude lost during dive recovery pullout is calculated by CAMPS.

Table 11.3.2-1 Altimeter Lag Implementation

```
DERATE VALUE

     DR = TAS*1.6878*SIN(DV/57.3)

DETERMINE ON OR OFF

     GO TO (100,200) IOOFF

SPC ON

     ILAG=DR*(.05-DR*3E-4)+.5

     IF( DV.LE.30 .AND.TAS.LE.350.) ILAG=40.

     IF( DV.LE.20. .AND.TAS.LE.460.)ILAG=40.

     IF( DV.LE.18.0 ) ILAG=TAS/30.+1.4*DV

     IF( DV.EQ.20..AND.TAS.GT.460) ILAG=ILAG+4

     IF( DV.EQ.30..AND.TAS.GE.450) ILAG=ILAG-4

     RETURN

SPC OFF F4E


     ILAG=DR*(1.9+DR*8E-4)+.5
```

11.4.2 Implementation

Using the number of G's, initial airspeed, and dive angle at pullout, CAMPS determines the pullout ramp path. The difference between the initial pullout altitude and the lowest ramp altitude is the net altitude lost. The Fortran code in Table 11.4.2-1 provides the algorithm used.

11.5 Calibrated Airspeed to True Airspeed

11.5.1 Description

Given the calibrated airspeed value, CAMPS converts to CAS true airspeed.

11.5.2 Implementation

CAMPS converts the temperature given in centigrade to kelvin and determines the current air pressure. Using the release altitude of the aircraft, CAMPS establishes the temperature ratio and transforms it to an ambient pressure ratio. The pilot's pressure difference is calculated and combined with the ambient pressure ratio to compute the true mach number. The true mach number is then transformed into true airspeed.

The Fortran code in Table 11.5.2-1 provides the algorithm used.

11.6 Angle of Attack

11.6.1 Description

CAMPS provides the aircraft (F-4E) angle of attack for each target.

11.6.2 Implementation

Assuming that density equals 11 lbs./cubic foot and that the speed of sound =661.5 mi/hr at current temperature, CAMPS uses the calibrated airspeed value to determine the mach number. Using the mach number value, dive angle, and altitude, CAMPS then generates the angle of attack.

The Fortran code in Table 11.6.2-1 provides the algorithm used.

11.7 Aim-Off Distance

11.7.1 Description

CAMPS has the capability to provide both the aim-off distance and the sight depression angle.

11.7.2 Implementation

CAMPS uses the dive angle and the release altitude to determine the flight path distance to the target/ground. Using this distance value CAMPS calculates the aim-off distance.

The Fortran code in Table 11.7.2-1 provides the algorithm used.

Table 11.4.2-1(a) Altitude Loss During Pull-Up Implementation

```
SPEED IN FT/SEC
      V=TAS*1.6878
NUMBER OF G'S FUNCTION OF DIVE ANGLE

      A=RDV
      G=NG-1+COS(A)
INITIALIZE LOSS AT 0
      AL= 0

CALCULATE ON RAMP

      DO 100 1=0,200,10
      G1=(1+(G-1)*I/200)
      A1=A
      G2=G1-COS(A)
      A=A-G2*(32.174/V)*.1
      A3=(A1+A)/2
      AL=AL+V*SIN(A3)*.1
      IF (A .LE. 0) GO TO 500
CONTINUE
```

Table 11.4.2-1(b) Altitude Loss During Pull-Up Implementation

```
CALCULATE FOR STEADY STATE



DO 200 I=210,6000,10
     G1=G
     A1=A
     G2=G1-COS(A)
     A=A-G2*(32.174/V)*.1
     A3=(A1+A)/2
     AL=AL+V*SIN(A3)*.1
     IF (A.LE.0) GO TO 500
CONTINUE
     ALTLOS=AL
```

Table 11.5.2-1 Calibrated Airspeed to True Airspeed Implementation

```
TEMP C TO K
     TK=IFTEMP+273
PILOT PRESSURE DIFFERENCE
     PPD=2116.217*((((CAS/661.4748)**2/5+1)**(3.5))-1)
TEMP RATIO
     TRATIO=1.-6.8756E-6*RALTM
AMBIENT PRESSURE RATIO
     APRATI=TRATIO**5.2559
AMBIENT PRESSURE
     APRES=APRATI*2116.2166
TRUE MACH NUMBER
     TM=SQRT(5*((PPD/APRES+1)**(.2857)-1))
TRUE AIR SPEED
     TAS=TM*SQRT(1.4*3089.776*TK)/1.688
```

Table 11.6.2-1 Angle of Attack Implementation

```
DENSITY
      D=1
SPEED OF SOUND AT TEMP
      A-661.5
MACH NUMBER
      RM=CAS/A
CALCULATE ANGLE OF ATTACK
      P=2116*D
      Q=.7*P*RM*RM
      C1=WOT*COS(DVR)/(530*Q)
      IAOA=(307.7*(C1-.044)-1.514E-4*CAS*CAS+.1363*CAS-29.476)+.5
```

Table 11.7.2-1 Aim-Off Distance Implementation

```
FLIGHT PATH DISTANCE TO GROUND IN FT
     DTG=RALTA/SIN(RDV)
SELECTED FUNCTION
     GO TO (100,200) IT
CALCULATE AIM OFF DISTANCE ADD
     DEPA=FLOAT(ISDEP)/1000.0
     XANG=3.14159-DEPA-RDV
     AOD=DTG*SIN(DEPA)/SIN(XANG)
     RETURN
CLACULATE DEPRESSION ANGLE ISDEP
     TEMP=(RALTA/TAN(RDV))-AOD
ISDEP=(1.5708-RDV-ATAN(TEMP/RALTA))*1000.0+.5
```

11.8 Pipper Placement

11.8.1 Description

CAMPS provides the pipper planning positions given the rollout speed and time of the final altitude or a specific altitude above the weapon release altitude.

11.8.2 Implementation

The pipper placement algorithms are based on the -34-1 nomograms, tables, and their derived interpolations. CAMPS uses one algorithm to determine pipper placement if the given time of release is the final or release altitude, and another algorithm if the given time of release is a specific number of feet above the release altitude. From the user input of time/altitude, CAMPS selects the appropriate algorithm and enters the given rollout speed and other data retrieved from files under angle of attack, aim-off distance, and calibrated/true airspeed data bases.

11.9 Target Density Altitude

11.9.1 Description

CAMPS provides target density altitude at the weapons release point. CAMPS determines the target density altitude from the following target parameters: user inputs of target elevation (MSL), altimeter setting over target, and target temperature.

11.9.2 Implementation

Target density altitude is a function of the current altitude and temperature. CAMPS converts the temperature given in centigrade to kelvin and establishes a temperature ratio at the current altitude. Using the temperature ratio, CAMPS computes the pressure at the current altitude. Using the pressure and altitude, CAMPS calculates the air density and establishes a density ratio which is used to formulate an equivalent temperature ratio (Temperature Density Ratio) to determine the target density altitude.

The Fortran code in Table 11.9.2-1 provides the algorithm used.

Table 11.9.2-1 Target Density Altitude Implementation

```
TEMP C TO K
     TK=ITTEMP+273
TEMP RATIO AT ALTITUDE
     TRATIO=1-(6.876E-6*TPA)
PRESSURE RATIO AT ALTITUDE
     PRATIO=TRATIO**5.2559
PRESSURE AT ALTITUDE
     PAALT=2116.22*PRATIO
DENSITY AT PRESSURE ALTITUDE
     DAALT=PAALT/(3089.78*TK)
DENSITY RATIO
     DRATIO=DAALT/.0023769
EQUIVALENT TEMP RATIO
     ETRATI=DRATIO**(1/4.2559)
TARGET DENSITY ALTITUDE
     TDA=(1-ETRATI)/6.876E-6
```

GLOSSARY OF TERMS

Coverage Template	Area of range of a threat site
Danger Template	Area of high risk in range of a threat site
Digitizer	Hardware which converts a physical quantity (i.e curves) into coded character form
Leg	In a flight route; the distance between two points
Mask	To block the range of threat sites
Pipper	The two-mil diameter dot in the center of the reticle of an optical sight
Probability of Damage	Probablity of sufficient damage to aircraft to cause abort of the mission
Standard Configuration Load	Particular distribution of specific aircraft stores
Strike Route	Same as flight path
Tactical Air Navigation System	Ground or ship-based system which enables an appropriately equipped aircraft to determine its range and bearing from the TACAN location

LIST OF ACRONYMS

AGL	Altitude Above Ground Level
CAMPS	Computer Aided Mission Planning System
CAS	Knots Calibrated Airspeed
CRT	Cathode Ray Tube
DMA	Defense Mapping Agency
ECM	Electronic Counter Measures
EOB	Electronic Order of Battle
FTR	Fuel Tank Release Point
IP	Initial Point
LCC	Lambert Conformal Conic
MOB	Main Operating Base
MSL	Altitude Above Mean Sea Level
NMI	Nautical Miles
PD	Probability of Damage
SCL	Standard Configuration Load
SPC	Static Pressure Correction
TACAN	Tactical Air Navigation System
UTM	Universal Transverse Mercator

82